

**REMARKS/ARGUMENTS**

This Amendment is in response to the Final Office Action dated August 1, 2003. Claims 1-18 are pending in the present application.

**35 U.S.C. § 103 Rejection**

The Examiner rejected claims 1-18 under 35 U.S.C. §103(a) as being unpatentable over Ponnekanti et al. (U.S. Patent No. 6,363,387). In so doing, the Examiner stated:

With respect to claim 1, Ponnekanti discloses utilizing a query processor to call a data manager and request the return of data from the set of data (database table storing data pages for query processing as data manager: col. 8, lines 6-20; and query processing calls to access data page table as a set of data: col. 13, lines 8-14); allowing the data manager to locate query-specified data on a stabilized data page (data page storing in the table with indexing: col. 9, lines 1-13 and col. 10, lines 28-32) and make a determination regarding the query-specified data (locating data page and maintaining the data page: col. 8, lines 38-46); utilizing the data manager to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page (manipulation or operations on index page or data page in the table: col. 16, lines 5-30; also see buffer manager for data page: col. 10, lines 46-67); and utilizing the query processor to retrieve the query-specified data from the buffer (col. 1, lines 40-50: extracting data page from database table where is storing data page; and col. 10, lines 52-67 and col. 11, lines 1-7).

Ponnekanti does not clearly disclose "to write the query-specified data on the stabilized of data page." However, Ponnekanti discloses the manipulating operations on the data page such as inserting, modifying and deleting (col. 16, lines 5-30, col. 2, lines 11-32 and col. 9, lines 45-60).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to utilize the manipulation operations for data page storing in the table (col. 16, lines 5-30) for reducing overhead as taught by Ponnekanti (col. 3, lines 22-30) because it would have made method for processing data page in the table being optimizing the query processing (col. 4, lines 35-44) and reducing the overhead of locking and increasing the concurrency, a particular performance advantage of the technique in querying data page (col. 10, lines 21-26) and this method would provide the index manager to locate the index and continue scanning the table at the next record row (col. 16, lines 20-40) in the fetching index-data environment.

Applicants respectfully disagree. The present invention is directed to a method and system for processing a database query on a set of data stored on a plurality of data pages in a database management system. In accordance with the present invention, a query processor calls a data manager to request the return of query-specified data from the set of data. The data manager locates the query-specified data on a stabilized data page and, if appropriate, writes the query-specified data to a buffer while maintaining the stabilization of the data page. The query processor then can retrieve the query-specified data from the buffer.

By writing the query-specified data to a buffer, instead of returning the data directly to the query processor, and maintaining the stabilization of the data page, the data manager operates more efficiently because it can continue to locate query-specified data without repeatedly restabilizing the data page. Query execution is more efficient because the number of times a data page is stabilized is minimized.

The present invention, as recited in claims 1 and 16, provide:

1. A method for processing a database query on a set of data stored on a plurality of data pages in a database management system, the method comprising the steps of:

- a) utilizing a query processor to call a data manager and request the return of data from the set of data;
- b) allowing the data manager to locate query-specified data on a stabilized data page and make a determination regarding the query-specified data;
- c) utilizing the data manager to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page; and
- d) utilizing the query processor to retrieve the query-specified data from the buffer.

16. A method for processing a database query on a set of data stored on a plurality of data pages in a database management system, the method comprising the steps of:

- a) utilizing a query processor to call a data manager and request the return of data from the set of data;
- b) allowing the data manager to locate query-specified data by:
  - b1) locating a data page containing query-specified data;
  - b2) stabilizing the data page; and

- b3) accessing the data page;
- c) utilizing the data manager to make a determination regarding the query-specified data and to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page; and
- d) utilizing the query processor to retrieve the query-specified data from the buffer.

Independent claims 6 and 11 are claims for a system and a computer readable medium containing program instructions, respectively, that are similar in scope to claim 1. Independent claims 17 and 18 are claims for a system and a computer readable medium containing program instructions, respectively, that are similar in scope to claim 16.

In contrast, Ponnekanti is directed to minimizing locking to optimize concurrency. In Ponnekanti, "locking is done after reading the data row, taking into account the status bits of data row and whether the data row qualifies." Col. 4, lines 36-43. During an index scan, Ponnekanti discloses deferred locking where the Index Manager returns RIDs to the Data Layer without acquiring locks on them during scans that have scan arguments on columns not in the index. The Data Layer subsequently qualifies the data row and determines whether locking is really necessary. Thus, the locking is done after the Data Layer reads the data row and determines that the data row qualifies. By determining whether an index or data row qualifies before applying the lock on the row, Ponnekanti reduces locking overhead and increases concurrency.

Applicants respectfully submit that Ponnekanti fails to teach or suggest the present invention, as recited in claims 1, 6, 11, 16, 17 and 18. In particular, Ponnekanti does not teach or suggest "utilizing the data manager to write the query-specified data on the stabilized data page to a buffer . . . while maintaining the stabilization of the data page" and "utilizing the query processor to retrieve the query-specified data from the buffer," as recited in claims 1, 6, 11, 16-18. In the present invention, during a table scan, the data manager stabilizes, i.e., latches, a data

page on which query-specified data is located, writes the query-specified data to a buffer (instead of returning it directly to the query processor) while maintaining the latch on the data page, and continues reading the rows on the data page. The query processor then fetches the data from the buffer. By writing the query-specified data to a buffer, query processing is more efficient because the number of page stabilizations is reduced.

Nothing in Ponnekanti teaches or suggests optimizing query processing *in this manner*. While Ponnekanti focuses on reducing the number of locks by qualifying a row before applying the lock, Ponnekanti makes no mention or suggestion of writing "the query-specified data to a buffer . . . while maintaining the stabilization of the data page" and retrieving "the query-specified data from the buffer," as recited in claims 1, 6, 11, and 16-18. The cited portions of Ponnekanti discuss database management systems in general (col. 1, lines 40-50; col. 2, lines 11-32), latching mechanisms in general (col. 10, line 52 to col. 11, line 7), and the index scan process (col. 16, lines 5-44). None of those portions mentions or suggests writing "the query-specified data to a buffer . . . while maintaining the stabilization of the data page" and retrieving "the query-specified data from the buffer."

The Examiner concedes that "Ponnekanti does not clearly disclose, 'to write the query-specified data on the stabilized of data page.'" The Examiner, however, contends that because Ponnekanti discloses performing "manipulating operations on the data page such as inserting, modifying and deleting," that it would have been obvious to one skilled in the art "to utilize the manipulation operations for data page storing in the table (col. 16, lines 5-30) for reducing overhead." Applicants respectfully disagree and submit that the ability to perform "manipulating operations on the data page" as disclosed in Ponnekanti neither teaches nor suggests writing "the query-specified data to a buffer . . . while maintaining the stabilization of the data page" and

retrieving "the query-specified data from the buffer," as recited in independent claims 1, 6, 11 and 16-18.

Applicants respectfully submit that Ponnekanti fails to teach or suggest the present invention, as recited in claims 1, 6, 11 and 16-18. Accordingly, claims 1, 6, 11 and 16-18 are allowable. Claims 2-5, 7-10 and 12-15 depend on claims 1, 6 and 11, respectively, and the above arguments apply with equal force. Therefore, Applicants respectfully submit that claims 2-5, 7-10 and 12-15 are also allowable.

#### Conclusion

In view of the foregoing, it is submitted that the claims 1-18 are allowable over the cited reference and are in condition for allowance. Applicants respectfully request reconsideration of the rejections to the claims.

Applicants believe that this application is in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,

**DRAFT**

September 30, 2003  
Date

\_\_\_\_\_  
Joyce Tom  
Sawyer Law Group LLP  
Attorneys for Applicant(s)  
Reg. No.: 48,681  
(650) 493-4540